

Ambience 4 Guide for Bulk Reporting Workload

1 Ambience 4 Overview

Ambience 4's default configuration for general workload as medium-large complex results with a lot of data integration. For bulk reporting workload, the Ambience 4 server requires a different configuration.

1.1 Ambience 4.6.5

Ambience 4.6.5 is the latest version and has a set of features that provide improvement for bulk report generation in areas of clean temps, reducing job logs. The enhancement are as listed below

	Description	Modules Impacted	Git Checksum
1	temp-deleter logic change to improve delete of temp files/ folder performance.	Vivace, Dacapo	012e7df
2	Optimisations for cleaning job as clean temp and clean job logs. The values for cleanTemp and cleanLogs can now be expressed with a unit as ms, s, m, h, d. Default unit is day	Arch ArchFramework Dacapo Choreographer	84daafd
3	Full implementation for no user logs. job log results are now returned along with the report output, they are not saved in the server. There is a config option: elixir.joblog.user-logs: true Add this and set this to false and no job logs will be created in the user folder. (it is true by default). Required for bulk reporting.	WebServices, Scheduler, AdhocDashboard, ArchFramework	1fa5adf
4	Force early close of InfoWatcher	ArchFramework	fa68358
5	VJournal Diagnostics tools		
6	Configurable temp files deletion		

Ambience 4.6.5 has been officially released on 1st November 2021.

2 Ambience 4 Configuration for managing temp files, user job logs and vjournal

2.1 Overview

The strategy adopted for bulk reporting in our test environment is to keep the vjournal and server temp folder size as small as possible. The configuration focuses on reducing the cleanTemp and cleanJob expiry time and compacting time interval of the repository filesystem and the vjournal.

Here are the summaries of the the steps needed

- Setup the “All in one Housekeeping” job and schedule it to run
- Set the cleanTemp and cleanLogs file values in Domain admin portal page
- Set the vivace compact frequency, file count value and set user job log

2.2 Configuration

The below settings is a reference configuration in our test environment . We set the housekeeping jobs to run every 15 minutes interval and 20 minutes for compacting vjournal file.

The configuration files for your reference is at

https://drive.google.com/file/d/1hzv5jO_fOij_io698Ja3_8F1knEqZ9nN/view?usp=sharing

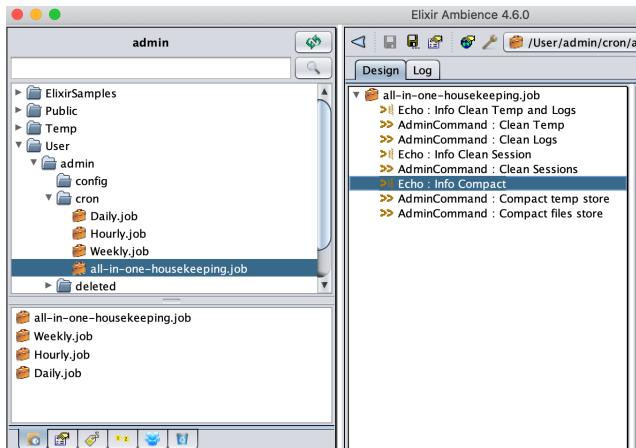
In the archive, there are three reference configuration files as

- all-in-one-housekeeping.job
- etc/application.conf
- etc/logback.xml

Users are advised to monitor and make the appropriate changes to the setting for their intended production environment.

2.2.1 All in one Housekeeping job

This new housekeeping job does the following, Clean temp files and logs, clean session and compact temp and file store. Compact files store need not be part of this housekeeping job, if there is no frequent change to the store.



You may deploy the “all-in-one-housekeeping.job” file using Ambience Designer. Place the job file in the Ambience repository admin user cron folder i.e. repository:/User/admin/cron as seen above.

Next go to the Ambience Web portal login as the admin user, Schedule (/elx/do/enopo/index.html#schedule). Create a new scheduled all-in-one-housekeeping job to run every 15 minutes. You may vary this value according to your environment.

Name	User	Previous	Next	Calendar
Daily	admin		2021-07-27 00:15:00 +08:00	Run Now
Hourly	admin	2021-07-26 12:00:00 +08:00	2021-07-26 13:00:00 +08:00	Run Now
Weekly	admin		2021-08-01 03:15:00 +08:00	Run Now
all-in-one-housekeeping.job	admin	2021-07-26 12:16:08 +08:00	2021-07-26 12:31:08 +08:00	Run Now

The screenshot shows the configuration page for the 'all-in-one-housekeeping.job'. The job is enabled and has a 'Duplicate' button. The job path is '/User/admin/cron/all-in-one-housekeeping.job'. The calendar is set to 'None' and the time zone is 'System Default Time Zone'. The job is scheduled to run with a regular interval of 15 minutes, repeating forever. The job has been triggered 14 times. The schedule starting date is 2021-07-26 at 00:00:00, and the schedule stopping date is 2021-07-26 at 00:00:00. The timeout is set to 60 seconds for both wait for start and wait for end. The misfire option is set to 'Refire the trigger immediately upon detection.' and the adjust repeat option is set to 'Adjust the repeat count as if successful.'

You may choose to disable the daily, hourly or weekly job once you have tested the system. But they did not have a significant impact on the test system.

2.2.2 Update the cleanTemp and cleanLogs value on domain admin page

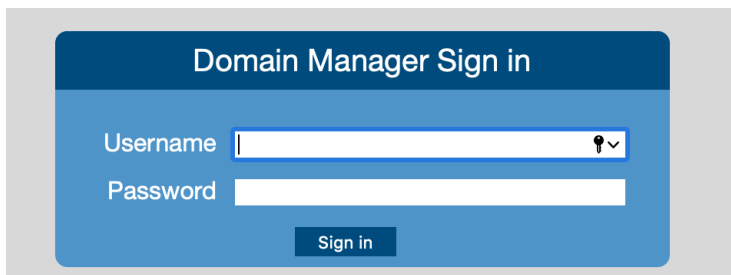
The values for cleanTemp and cleanLogs can now be expressed with a unit:

Allowed units are:

- **ms** = milliseconds
- **s** = seconds
- **m** = minutes
- **h** = hour
- **d** = day (default)

Set the cleanTemp and cleanLog configuration value using web portal domain configuration. Do not use the Designer configuration panel as this will overwrite the values set.

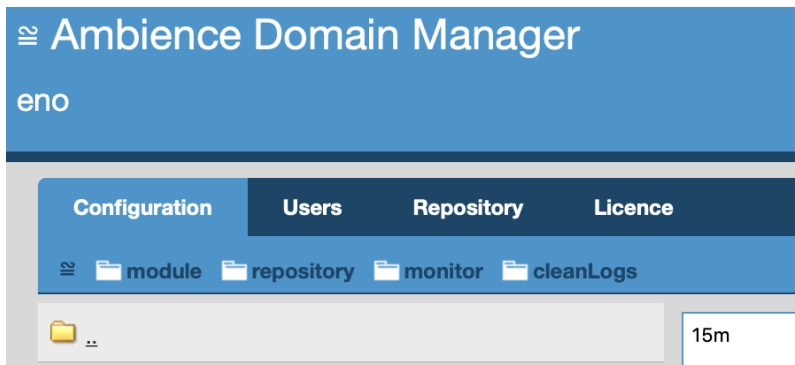
Login into domain admin web page (/elx/dm/eno/index.html#configuration) .



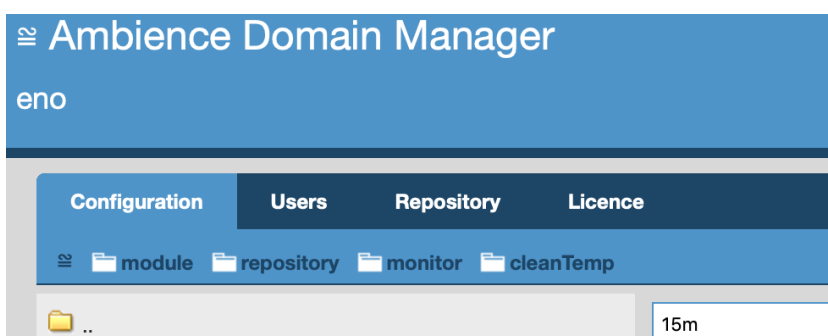
Go to the configuration tab and navigate to “module/repository/monitor/” node.

Make two changes to the Repository cleanTemp and cleanLogs time i.e. 15 m (15 minutes) and save the changes as seen below

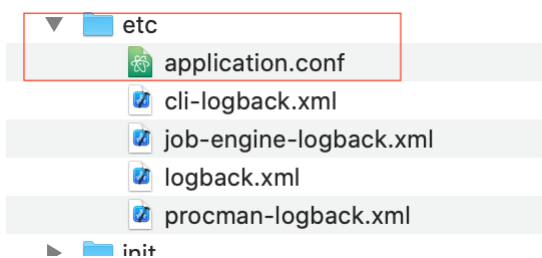
i) cleanLogs - module/repository/monitor/cleanLogs



ii) cleanTemp - module/repository/monitor/cleanTemp



2.2.3 Update the vivace compact frequency, file count and user-logs value in the “application.conf” Ambience configuration file in “etc” folder



The values used in the test environment are

- elixir.vivace.compact-frequency=20 minutes
- elixir.vivace.backup-count=6
- elixir.joblog.user-logs=false

```
akka.remote.netty.tcp.port = ${seed-port}
akka.cluster.roles = ["vivace", "seed"]

elixir.vivace.compact-frequency=20 minutes
elixir.vivace.backup-count=6
elixir.joblog.user-logs=false

elixir {
  root = ${root-dir}
```

When “elixir.joblog.user-logs” is set to false, job log results are returned along with the report output, they are not saved in the server. This prevents the “data/dacapo/eno/repository-temp/” folder size from growing rapidly. You may monitor to “data/dacapo/eno/repository-temp/” folder after the housekeeping job is done to ensure that this parameter is working,

2.2.4 Other optimization to improve performance

a) Disable indexer in application.conf as it is not needed for bulk report generation.

```
indexer {
  dir = ${root-dir}/data/indexer"
  initial-delay = 5 minutes
  update-interval = 15 days
  max-results = 1000
  enabled = false
  indexers = ["ds", "pdf", "rml", "text"]
}
```

b) Reduce Ambience logging

You may reduce the ambience logging by setting the amount of information to be logged out into the file as in the logback.xml configuration to warn instead.

```
<logger name="com.elixirtech" level="warn" />
<logger name="com.elixirtech.ui.SystemStatus" level="warn" />
<logger name="com.elixirtech.vivace" level="warn" />
```

To see server activities only here is new logger entry

```
<logger name="com.elixirtech" level="warn" />
<logger name="Activity" level="info" />
<logger name="com.elixirtech.ui.SystemStatus" level="warn" />
<logger name="com.elixirtech.vivace" level="warn" />
```

This only logs activity logs.

You may do the same to job-engine-logback.xml change com.elixirtech level to warn

```
<logger name="com.elixirtech" level="warn" additivity="false">
```

This could cut the logs to about half the size as compared to the default configuration.

For more information on logback, you may refer to <http://logback.qos.ch/documentation.html> .

c) Configurable .temp files deletion

This feature enables you to configure to avoid output of temp files & is handy when you're generating a large amount of reports/files. It can be enabled by setting it to "true" in "/etc/application.conf" as shown below.

```
# When generating a lot of temp files, eg. with a batch of a million reports
# set this to true to avoid generating lots of temp files that will take time to clean up later
job.fast-temp-cleaner.enabled = true
```

3) Ambience 4 sizing and approach for bulk reporting in Test System

3.1) Test system size

The recommendation sizing is 1 core per process i.e 1 core for Ambience Server and per core for Jobengine. However the number of cores is dependent on the vendor of CPU and the server environment. And 1- 2 GB of Ram per process. For example an Ambience and 4 Jobengine would need about 8 GB ram to be allocated.

In our testing system for Ambience server with 4 Job Engines running on vmware , the optimal configuration is to have at least 4 - 6 VCPUs assigned to the system and 8 GB of Ram. As having more threads will help in the system processes i.e. OS to run and Java runtime to perform its GC job and for the Ambience server to run and reclaim resources.

3.2) Java - Server and Job Engine configuration

The Java configuration can be set up on the Ambience Server with a GC flag “**-XX:+UseG1GC**”,



```
#!/bin/sh
```

```
cd ..  
OPTIONS="-XX:+UseG1GC -Dvisualvm.display.name=AmbienceServer -Delixir.home=. -Djetty.home=jetty  
-Djetty.base=jetty/elx-base -Djava.awt.headless=true -Dlogback.configurationFile=etc/logback.xml"
```

```
java $OPTIONS -jar elx-stub.jar org.eclipse.jetty.start.Main
```

The same should be set up for job engines in application.conf. This setting directly impacts on the actual report generation.

Below is an example from our test environment

```
ambience {
  init-modules = ${root-dir}/init"
  init-domains = [${domain}]
  start-domains = [${domain}]
  stop-jobengines-on-shutdown: true

  local-job-engines = [
    {
      name = "first"
      domain = ${domain}
      # vm-args is optional, you can include strings here such as
      # ["-Xmx6G", "-server" ] or leave it out completely
      vm-args = ["-XX:+UseG1GC"]
    }
    {
      name = "second"
      domain = ${domain}
      vm-args = ["-XX:+UseG1GC"]
    }
    {
      name = "third"
      domain = ${domain}
      vm-args = ["-XX:+UseG1GC"]
    }
    {
      name = "fourth"
      domain = ${domain}
      vm-args = ["-XX:+UseG1GC"]
    }
  ]
}
```

```
local-job-engines = [
  {
    name = "first"
    domain = ${domain}
    # vm-args is optional, you can include strings here such as
    # ["-Xmx6G", "-server" ] or leave it out completely
    vm-args = ["-XX:+UseG1GC"]
  }
]
```

3.3) Run in split batches rather than in a single run and spread the workload across a series of Ambience server

It is recommended to break up the big report generation workload into a smaller set of workload to be spread over a range of Ambience Servers (scaling out) instead of running a large single workload on one Ambience instance. And the workload has to properly load balanced across servers.

4) Other Future Considerations

However for bulk reporting the Repertoire runtime or A2021 is a better choice. For best results a custom engine is possible - this would save loading the report multiple times, if it is the same report with just varying data.

Another option would be to generate one big report (lots of pages) and then split those pages later. This would provide better performance with A4.